Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 49604, 11 pages
doi:10.1155/2007/49604

Research Article

Identifying Opportunities for Exploiting Cross-Layer Interactions in Adaptive Wireless Systems

Troy Weingart, Douglas C. Sicker, and Dirk Grunwald

Department of Computer Science, University of Colorado, 430 UCB, Boulder, CO 80309-0430, USA

Received 31 December 2006; Revised 15 April 2007; Accepted 2 July 2007

Recommended by Zhu Han

The flexibility of cognitive and software-defined radio heralds an opportunity for researchers to reexamine how network protocol layers operate with respect to providing quality of service aware transmission among wireless nodes. This opportunity is enhanced by the continued development of spectrally responsive devices–ones that can detect and respond to changes in the radio frequency environment. Present wireless network protocols define reliability and other performance-related tasks narrowly within layers. For example, the frame size employed on 802.11 can substantially influence the throughput, delay, and jitter experienced by an application, but there is no simple way to adapt this parameter. Furthermore, while the data link layer of 802.11 provides error detection capabilities across a link, it does not specify additional features, such as forward error correction schemes, nor does it provide a means for throttling retransmissions at the transport layer (currently, the data link and transport layer can function counterproductively with respect to reliability). This paper presents an analysis of the interaction of physical, data link, and network layer parameters with respect to throughput, bit error rate, delay, and jitter. The goal of this analysis is to identify opportunities where system designers might exploit cross-layer interactions to improve the performance of Voice over IP (VoIP), instant messaging (IM), and file transfer applications.

1. INTRODUCTION

The flexibility of cognitive and software-defined radios presents an opportunity for researchers to reexamine how network protocol layers operate with respect to providing quality of service aware transmission among wireless nodes. This opportunity is enhanced by the continued development of spectrally responsive devices—ones that can detect and respond to changes in the radio frequency environment. Present wireless network protocols define reliability and other performance-related tasks narrowly within layers. For example, the frame size employed on 802.11 can substantially influence the throughput, delay, and jitter experienced by an application, but there is no simple way to adapt this parameter. Furthermore, while the data link layer of 802.11 provides error detection capabilities across a link, it does not specify additional features, such as forward error correction schemes, nor does it provide a means for throttling retransmissions at the transport layer (currently the data link and transport layer can function counterproductively with respect to reliability).

This paper presents an analysis of the interaction of physical, data link and network layer parameters with respect to throughput, bit error rate, delay, and jitter. We look specifically at (1) power, (2) bitrate, (3) forward error correction, (4) frame size, (5) automatic repeat request, and (6) selective queueing. The goal of this analysis is to identify key opportunities for system designers to exploit cross-layer interactions improving the performance of Voice over IP (VoIP), instant messaging (IM), or file transfer applications. This research utilizes both simulation and system platforms in its analysis. For simulation purposes, we employed the OPNET network simulation environment. In developing an actual radio, we reverse engineered a commercial off-the-shelf (COTS) product to create adaptive data link and routing layers. The extensions to the COTS platform, what we refer to as SoftMAC and MultiMAC, served as an experimental framework for the implementation and evaluation of our simulation results.

Our analysis details those parametric interactions that significantly impact wireless performance. This analysis would then be used to inform the design, implementation, and evaluation of adaptive wireless networking algorithms. These algorithms will be able to dynamically adjust their behavior, thereby improving wireless performance in response to a changing radio frequency (RF) environment. Our work demonstrates that even conservative adaptation of the physical, data link, and network layers will provide significant enhancement to the performance of the system, and that adaptation can strongly influence the ability of the system to meet quality of service (QoS) requirements.

2. BACKGROUND

Cognitive and software-defined radios (C/SDRs) are one of the most promising of the emerging technologies addressing spectrum use, wireless performance, reconfigurability, and interoperability. Many research groups are actively pursuing projects that exploit these opportunities [1–8]. Much of the research and development in software-defined radio has been characterized by the building of systems to solve specific problems. In the United States Military, efforts are focused on waveform portability and radio interoperability. Research done at the University of Kansas was centered on mobile and rapidly deployable disaster response communication. Vanu Bose's thesis focused on solving problems associated with moving traditionally analog or custom ASIC components and processes to a general-purpose processor. Most of the problems in software radio lie in the development of more capable hardware and interoperable software frameworks. Software radio research, although very active, has begun to give way to the recent popularity of cognitive radio.

In the radio space, Moore's law has also provided momentum in transitioning from special-purpose inflexible hardware and firmware to mutable virtual radios [9]. These virtual radios or SDRs make use of general-purpose processors and software to accomplish what used to be done with specialized hardware. One can envision an advanced radio network that dynamically allocates and reallocates spectrum or dynamically reconfigures itself in response to changes in policy and environmental conditions. This level of flexibility in a radio platform not only allows us to tackle the problem of spectrum utilization, but also serves as a highly capable platform from which one can exploit cross-layer interactions.

Much of the resistance in the cellular industry to emerging third generation technologies stems from the huge cost involved when removing and replacing existing infrastructure with technology that can support the new standard [10]. On the other hand, a C/SDR platform may be able to adapt to new standards by downloading and installing new software. The flexibility inherent in the C/SDR also allows it to adapt to changes in policy. The recent focus on homeland security, in light of the ineffective use of the emergency bands, has given impetus to many of the scenarios that illustrate the promise of C/SDR. One could imagine a government agency implementing a change to local spectrum policy in response to a disaster. Updates to policy, when acted upon by a C/SDR network, could affect reallocation of spectrum to support increased demand during the emergency. The C/SDR is also ideally suited for system-level cross-layer networking research. Additionally, a C/SDR could be used for research and experimentation with spatially aware applications, adaptive routing, cognitive media access control (MAC) layers, and mutable physical layers.

At the top end of the software radio taxonomy are radios that incorporate computational intelligence. These "cognitive" radios will be able to sense, learn, and act in response to changes in their environment [7]. The ultimate cognitive radio will be able to autonomously negotiate and propose entirely new optimized protocols for use in the networking environment. Although provocative, current radio technology is nowhere near mature enough to realize systems with these capabilities. At the lower end of the C/SDR spectrum are radios that are functionally equivalent to their analog predecessors, although the newer radio's functionality has been implemented on field programmable gate arrays (FPGA), digital signal processors (DSP), or general-purpose processors (GPP). By in large, the systems of today were built to solve a domain-specific problem, or are focused on tightly coupled manipulations of the lower layers of the protocol stack. Regardless of where one's research interests lie, there are a host of technical challenges to overcome.

The focus of this paper is on understanding how varying parameters at the physical, data link and network layers can affect the performance and reliability of a wireless system. Understanding these effects is a critical first step in the development and implementation of an algorithm for cognitive radio. In addition, once the performance implications of varying these parameters is understood, one must also consider a host of implications that arise when one alters such parameters. This includes decisions on when and how to change configurations, how these changes are communicated, and how much time can be spent calculating the next configuration. While these are all important questions, we focus on understanding the impact of varying a C/SR's settings on its performance.

3. RELATED WORK

Research in the area of cross-layer optimization for wireless systems has been an area of considerable focus in recent years. Others have also spent a considerable amount of time and effort investigating cognitive radios. However, the potential of improving the performance of a wireless system by combining cross-layer optimization with cognitive systems is just emerging as a research area.

Much of the work in the area of cross-layer optimization focuses on enhancing throughput, quality of service (QoS), and energy consumption [11–13]. These cross-layer optimizations tend to focus on two layers of the protocol stack with the goal of enhancing a specific performance measure. As such, they do not consider multifactor variation nor do they consider effects of this variation on inelastic applications, such as Voice over Internet Protocol (VoIP). Kawadia and Kumar present an interesting critique of cross-layer design in [14]. They warn that cross-layer optimization presents both advantages and dangers. The dangers they discuss include the potential for (1) spaghetti design, (2) proliferation problems and, (3) dependency issues. Such cautions (and others that we will identify) are easily overlooked in the hopes of gaining sometimes marginal performance improvements. Therefore, understanding the significance of the potential improvements is an important step to consider.

Given that the interactions among a set of parameters are determined, the next step is determining the significance of these interactions. In other words, those interactions provide the best response in a given situation. Vadde et al. have applied response surface methodology and design of experiments (DOE) techniques to determine the factors that impact the performance of mobile ad hoc networks (MANETs) [15–17]. Their research considers routing protocols, QoS architectures, media access control (MAC) protocols, mobility models, and offered load as input factors and throughput and latency as response factors. Their analysis demonstrates the usefulness of these techniques and shows where certain input factors can outperform others within a MANET.

Haykin provides a thorough overview of cognitive radios and describe the basic capabilities that a "smart" wireless device might offer [18]. Others describe techniques for applying C/SDRs to improving the coordinated use of spectrum [19, 20]. Sahai et al. describes some of the physical layer limits and limitations of cognitive radios, including the difficulties associated with determining whether or not a radio frequency band is occupied [21]. Nishra has implemented a test bed for evaluating the physical and data link layers of such networks [22]. Additionally, Thomas describes the basic concept of a C/SDR network and provides a case study to illustrate how such a network might operate [23]. It is also worth noting that the standards communities are focusing on cognitive radios. The IEEE 802.22 group is developing a wireless standard for the use of cognitive radios to utilize spectrum in geographically separated and vacant TV bands [24]. Also in the IEEE, the P.1900 workgroup is examining the general issue of spectrum management in next generation radio networks.

4. EXPERIMENTAL DESIGN

In this section, we describe the design and implementation of the simulation and experimental platforms. We also introduce design of experiments (DOE), a technique that we employ in the identification of those parameters that significantly impact performance.

4.1. Simulation tool

In order to determine the validity of our approach, we conducted our preliminary research on a simulation platform. Upon considering the potential complexity of a cognitive network composed of many nodes, we decided to begin by evaluating a simple network. The simulation itself consisted of two nodes communicating in the presence of an active noise source (e.g., a noncooperative node on a different network, or a radio frequency jammer). We used OPNET Modeler to simulate the effects of changing communication parameters in order to determine where best to employ cross-layer optimization [25]. The simulation suite provides a rich and readily extendable network modeling environment. While OPNET provides a wireless networking module for the data link layer, to obtain the flexibility that was required for interactions spanning protocol layers, we found it necessary to develop our own data link module. This module allows adaptation of the parameters affecting cross-layer interaction on a per-packet basis.

The simulation platform uses an additive white Gaussian noise (AWGN) model to simulate the effects of environmental noise. The jammer in our simulation emits RF energy in bursts of varying duration and interarrival using OPNET's 802.11 physical layer.

4.2. Platform

Much of the work in developing the platform for this research has already been completed. This research relies on the use of COTS products with C/SDR extensions. The extensions to the COTS platform, SoftMAC and MultiMAC, serve as the experimental framework for implementing and evaluating the results of the simulation. The following sections describe each component of the platform used in the research.

4.2.1. SoftMAC

This system was built to provide a flexible environment for experimenting with MAC protocols in the wireless domain. The ability to cheaply create, modify, and conduct system-level experimentation with hardware is often a goal of many research projects. However, many of these projects ultimately fail due to the cost, time, and effort involved in deploying a large-scale experimental platform. The SoftMAC platform fills this need. It uses a commodity 802.11b/g/a networking card with a chipset manufactured by the Atheros Corporation to build a software radio with predefined physical layers but a flexible MAC layer. Internally, the Atheros chipset provides considerable flexibility over the format of the transmitted packets, network drivers do not generally expose this flexibility. By reverse-engineering many of those controls, SoftMAC provides a driver that allows extensive control over the MAC layer while still allowing use of the waveforms defined by the underlying 802.11b/g/a physical layers.

4.2.2. MultiMAC

This system is intended to extend the basic SoftMAC environment to tackle problems in the areas of dynamic spectrum allocation and cognitive/software-defined radio. It builds upon the functionality in the SoftMAC platform with some specific features in mind. First, MultiMAC allows multiple MAC layers to coexist in the network stack with minimal switching impact. Second, it allows one to dynamically reconfigure the MAC and physical layers on a per-packet basis either from logic running as part of MultiMAC or from a user-level process. Finally, by leveraging these capabilities MultiMAC allows intelligent reconfiguration of the MAC and physical layers; thus achieving a cognitive MAC. The cognitive MAC layer couples efficient reconfiguration afforded by MultiMAC with computational intelligence. This combination allows the engine to make smart decisions about which MAC layer should be used and which physical layer properties should be set.

Table 1 lists parameters that might be available to a MultiMAC cognitive process running on the platform. This

Table 1: A potential set of mutable parameters.

Parameter	Datatype
Route	Enum
Frame size	Integer
Forward error correction	Enum
Automatic repeat request	Boolean
Encryption	Boolean
Media access protocol	Enum
Channel	Integer
Modulation	Enum
Bitrate	Enum
Antenna configuration	Integer
Transmit power	Float

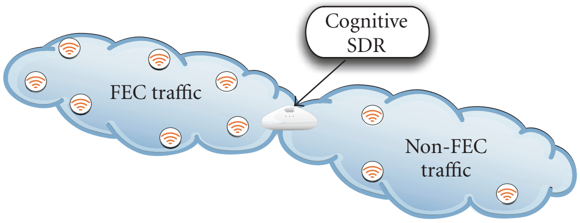

Figure 1: Dynamic MAC layer bridging using C/SDR.

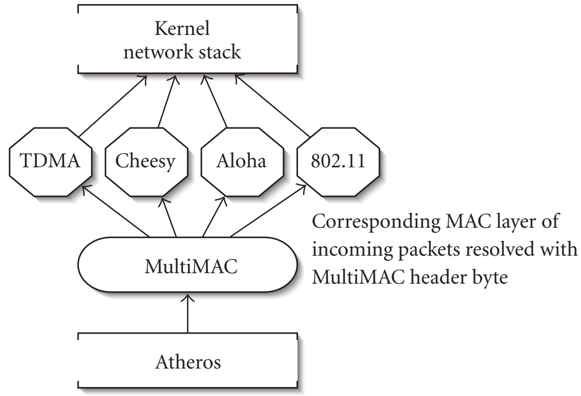

Figure 2: MultiMAC assigns received frames to the MAC layer that can decode them.

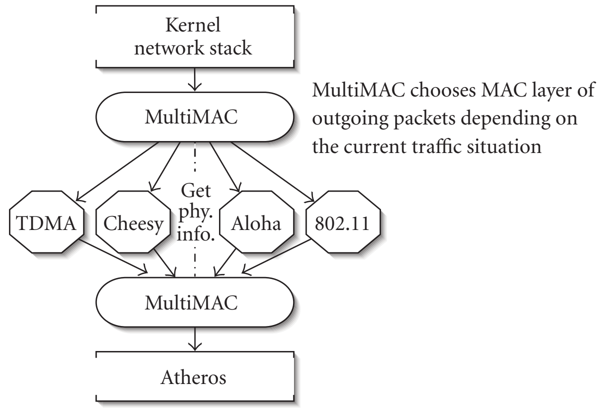

Figure 3: On the sending side, MultiMAC uses environmental stimuli to determine the path of a packet.

smart radio would use these parameters to effect changes in reliability due to its assessment of performance metrics and environmental inputs. The basic mechanism provided by MultiMAC can also be used to implement specific fixed MAC protocols. For example, one platform referenced in our work couples commodity 802.11 network cards with the Phocus phase array antenna. The directional phase array antenna will be able to use dynamic beam-forming to spatially create separate MAC zones (see Figure 1). One segment could use a forward error corrected (FEC) MAC layer while the other zone could use a MAC which does not have FEC enabled. The framework allows us to instantly transit from an FEC to a non-FEC MAC. Also, MultiMAC offers the ability to dynamically change properties while still decoding frames sent from previous configurations. Individual MAC variants will be used by MultiMAC for decoding their respective incoming frames as well as encoding outgoing frames with a MAC best suited for network conditions. This process is both completely transparent and highly adaptive in operation. MAC layers can be changed on the fly without interrupting radio service or dropping frames during the transition. When a decoded frame arrives, the appropriate MAC layer must "*claim*" and decode the frame. Once a packet is handed over to its corresponding MAC layer implementation, decoding happens the same way as in an unmodified network stack (see Figure 2). A mirrored procedure takes place on the encoding side; the process is more complex due to timing constraints imposed by the MAC layer protocol. (see Figure 3). The individual policies used to select an outgoing MAC protocol rely on cross-layer feedback from the physical and network layers. MultiMAC maintains a connection to the status and diagnosis API of the wireless chipset. By pulling out network status information such as the *time to transmit frames*, *queue lengths*, and *bit-errored frames*, MultiMAC policies can determine the appropriate MAC for the specified goal. (See [26] for details on SoftMAC and MultiMAC.)

4.3. Design of experiments

Design of experiments (DOE) is an approach for determining cause and effect relationships within an experiment [27]. Historically, this technique has been applied with great success in the process and materials industries. Later, we show that it also can be applied with success in the wireless domain. DOE provides a structured method for understanding the relationships among input and output variables. By systematically varying all the input factors, DOE allows researchers to identify the existence of interactions among these inputs and their impact on output factors. It allows researchers to determine what factors most influence an experiment and, moreover, determine the interaction among a group of input factors. In this paper, we rely on DOE to quantify the influence of single and multifactorial inputs illustrating how parameters across multiple layers might

Table 2: The set of mutable parameters.

Parameter	Settings	Layer
Automatic repeat request (ARQ)	Off/on	MAC
Frame size	2048, 9216, 18432 bits	MAC
Forward error correction	Off/on	MAC
Bitrate	1, 2, 5.5, 11 Mbs	Physical
Transmit power	5, 32, 100 mW	Physical
Selective queuing	Off/on	Network

improve (or degrade) the performance of a wireless system. This technique relies on the analysis of variance (ANOVA) statistical method to provide an assessment of the significance of the test results.

5. RESULTS AND DISCUSSION

The following section reports the results of our simulation work. The experimental trials were designed to cover a range of traffic sent between nodes in the presence of a noise source. FTP and VoIP traffic were selected due to their distinct tolerances for latency, jitter, throughput, and bit loss. The parameters that we examined included ARQ, frame size, bitrate, transmit power, FEC, and selective queueing (as described in Table 2). Stop-and-wait is the type of ARQ used, wherein the sending node will stop transmitting until it has received an acknowledgment from the receiver (or it times out; in which case the sender will retransmit the frame). When selective queuing is enabled, high priority frames (in our case, VoIP frames) are moved to the head of the transmit queue.

Each of the simulation trials was analyzed across the levels of the parameters and general trends were highlighted. For example, we looked at the average jitter, latency, bit loss, and throughput performance of FEC across all combinations of the other parameters. We then analyzed each of the traffic types using DOE techniques.

5.1. *Simulation configuration*

Figure 4 shows the physical layout of the two communicating nodes in relationship to the noise source. The uncooperative (or jamming) node is emitting noise in a Poisson distribution centered around an interarrival time of 0.05 seconds and a burst length of 1024 bits. The physical layout of the nodes and noise source as well as the power of the noise source is fixed across all of the trials; however, the duration and interarrival of the jamming bursts do vary. In our preliminary research, we examined a broad range of noise settings including different duration bursts, interarrival times, and power levels. We settled on a setting that provided appreciable interference without overwhelming the communicating nodes.

Each of the trials examines the performance of the experimental system at each of the potential parametric settings. Table 3 is a list of the metrics by which we evaluate each mutation of the settings. One can independently look at the performance of any of the parameters (alone or in combination with other parameters) against any one of the metrics used

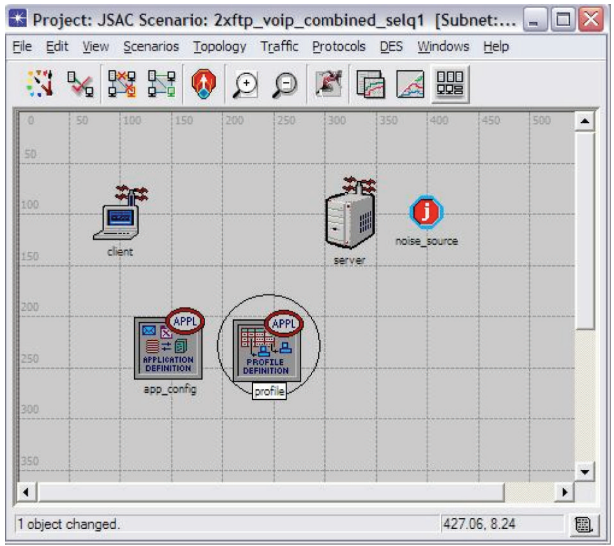

Figure 4: Experimental layout in OPNET.

Table 3: The set of metrics.

Metric	Units
Bit loss	Percent
Latency	Seconds
Jitter	Seconds
Throughput	bps

to evaluate the system. Next, we examine each of the trials in turn.

5.2. *File transfer protocol (FTP)*

5.2.1. *Experimental setup*

This first scenario was designed to isolate FTP traffic from the client to the server in the presence of a noise source. Here, FTP traffic is modeled using OPNET's client and server FTP traffic profiles. A 5 MB file is transfered from the client node to the server. In these experiments, we focused on optimizing throughput.

5.2.2. *General trends*

This analysis was done by fixing a parameter and reporting the average performance of that action across all permutations of the other parameters. For example, in order to investigate the general effect of ARQ on bit loss we started by first disabling ARQ and then running through all the permutations of the other parameters. This is followed by enabling ARQ and rerunning the simulation set. We then compare the average effect of enabling and disabling ARQ on bit loss. Figure 5 shows the performance of each of the parameter settings on throughput. One can see from the chart that increasing bitrate and frame size have a significant effect on throughput. This chart shows the average effect of changing a parameter; it does not show the overall best- or worst-case

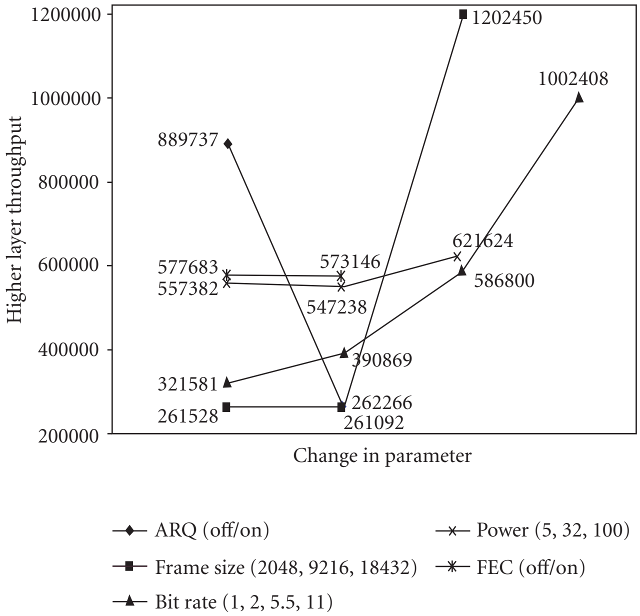

Figure 5: FTP throughput versus average effect of a parameter.

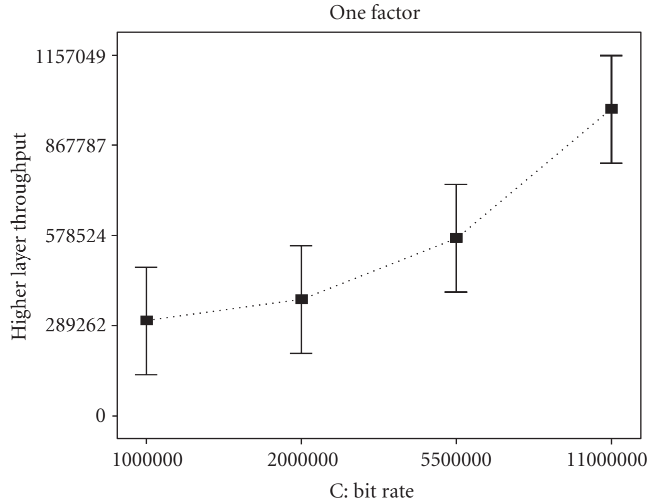

Figure 6: Analysis of bitrate's effect on FTP throughput.

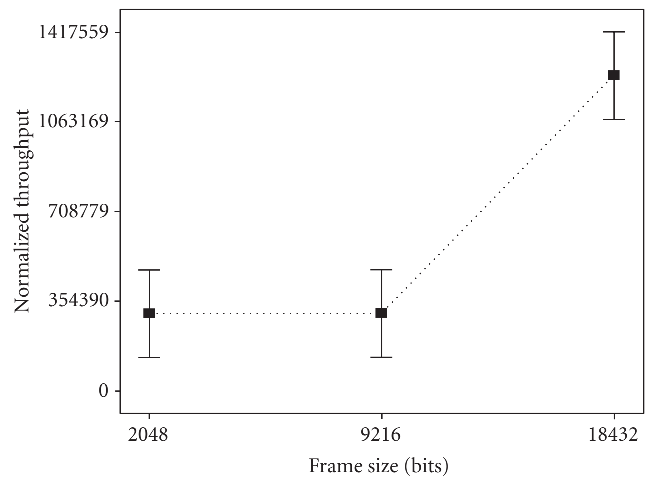

Figure 7: Analysis of frame size's effect on FTP throughput.

parameter settings. For FTP, the worst selection of parameters yielded a throughput of 10 Kbps and the best slightly above 9.5 Mbs.

5.2.3. Design of experiments (DOE)

It is appropriate to begin this section with some supporting information on the use of DOE. DOE is ideally suited to help us identify those cross-layer interactions that are statistically significant. DOE provides a design methodology and set of statistical tools for setting up and running experimental trials in a manner that allows one to identify those factors that significantly impact what you are measuring. In our case, DOE serves to identify both intra- and cross-layer interactions that effect the response of interest. The core statistical process at work in DOE is the calcuation of the *F-test*. This test compares the variance among the treatment means versus the variance of the individuals within the specific treatments. Another way of looking at F is as a ratio of signal to noise.

DOE analysis of the *main factor effects* of the parameteric change on throughput yeilded some interesting results. We first considered the main effects on FTP traffic (*main effects* can be defined as the change in response caused by altering a single factor). We found bitrate and frame size (as shown in Figures 6 and 7) to most improve throughput and ARQ to have a detrimental influence on throughput. Note that the DOE Y axis provides a normalized scale and therefore we do not discuss the quantitative results of these experiments; rather we focus on the trends and the interactions among the parameters. As expected, large frames and/or high bitrate improve throughput. Additionally, we found power and forward error correction to have little or no effect. Power does have a significant impact when we increase power and or frequency at the noise source; however, our general intention is to minimize the use of power because of the detrimental impact on neighboring nodes.

Next, we examined the two-factor effects on FTP throughput. As shown in Figure 8, frame size and data rate show a strong synergistic effect on improving throughput. Note that the top two lines both contain *least significant difference* bars that do not overlap, which indicates that the result is significant. Sending large packets at a high rate should improve throughput. The significance here is the magnitude of the improvement. Again, given the noise level, power provided little improvement in the achieved throughput.

5.3. Voice over IP (VoIP)

5.3.1. Experimental setup

This next scenario was designed to isolate VoIP traffic between the client and the server in the presence of a noise source. Here, VoIP traffic is modeled using OPNET's IP Telephony Model. We designed this experiment to demonstrate

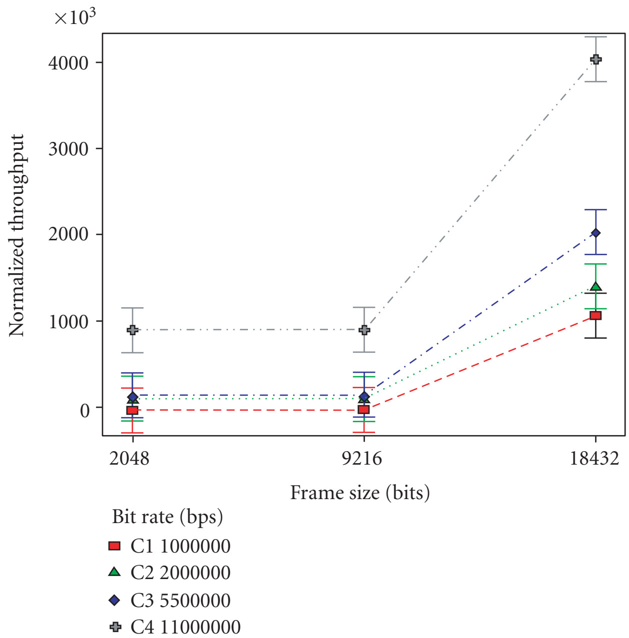

Figure 8: Analysis of frame size and data rate's effect on FTP throughput.

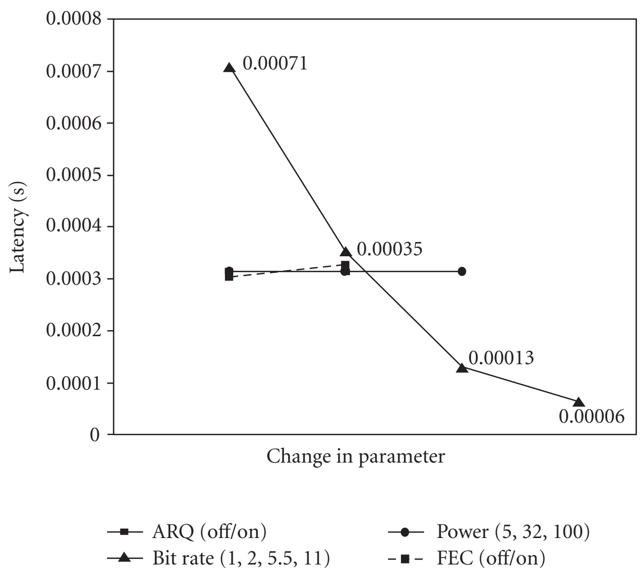

Figure 9: VoIP latency versus average affect of a parameter.

the behavior of VoIP on a lightly loaded network (later we look at VoIP performance on a heavily loaded network).

5.3.2. General trends

Figure 9 shows the performance of each of the parameter settings on latency (we also examined *jitter* and later indicated where it negatively and positively impacted VoIP traffic). One can see that increasing bitrate has the most significant effect on latency. Again, this chart shows the average effect of a parameter setting; it does not show the best- or worst-case

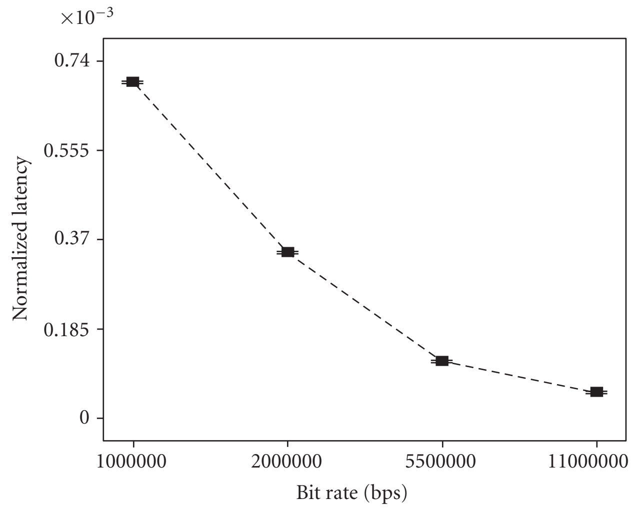

Figure 10: Analysis of bitrate's effect on VoIP latency.

configuration. The worst-case configuration yielded an average latency of 0.0073 seconds and the best 0.00006 seconds. Since this trial was conducted on a lightly loaded network, the other parameters (namely, ARQ, frame size, FEC, and power) had little impact on latency in the average case.

5.3.3. Design of experiments (DOE)

As shown in Figure 10, the DOE analysis confirmed the impact of bitrate on VoIP latency. The DOE analysis also confirmed that none of the other parameters had a significant main or multifactor effect on latency.

One can see from the chart that increasing bitrate has the most significant effect on latency. DOE multifactor analysis did not yield any statistically significant results.

5.4. FTP/VOIP combined

5.4.1. Experimental setup

Our goal in combining the two traffic types was to see how the parameter changes effect our metrics when the traffic profiles differ and overlap. This trial models the sending of a file from the client to the server during a VoIP call. Both of the traffic sources in this simulation are modeled with OPNET's constant bitrate sources. For this trial, we developed a selective queuing mechanism above our MAC layer. Selective queuing gives priority to VoIP frames by moving them to the front of the transmit queue.

5.4.2. General trends

We found a number of interesting results in this set of experiments, as shown in Figures 11 and 12. By selectively queueing VoIP frames, we were able to drastically improve latency while minimally impacting FTP throughput. Furthermore, both bitrate and power had a beneficial impact on both VoIP and FTP traffics. One can see from the chart that turning selective queuing on has a significant effect on latency

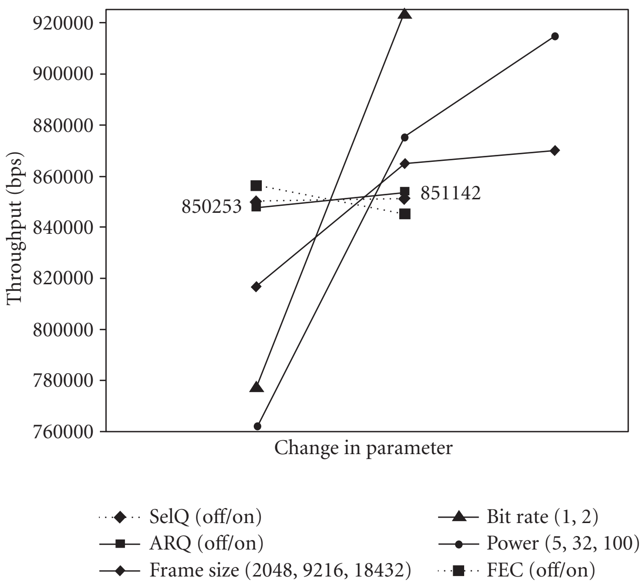

Figure 11: FTP throughput versus average affect of a parameter.

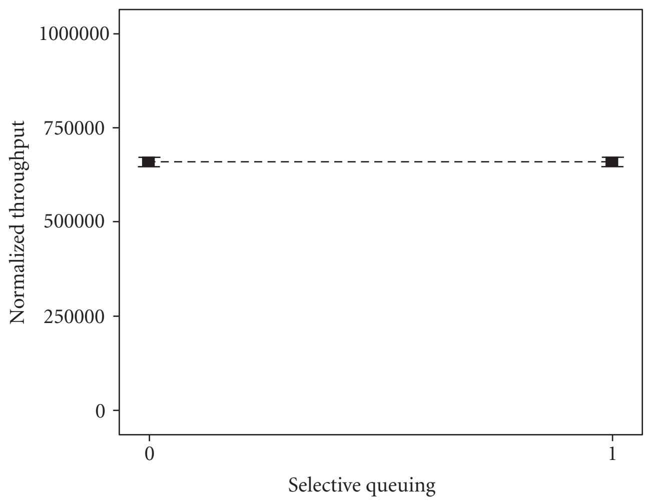

Figure 13: Analysis of selective queuing's impact on FTP throughput.

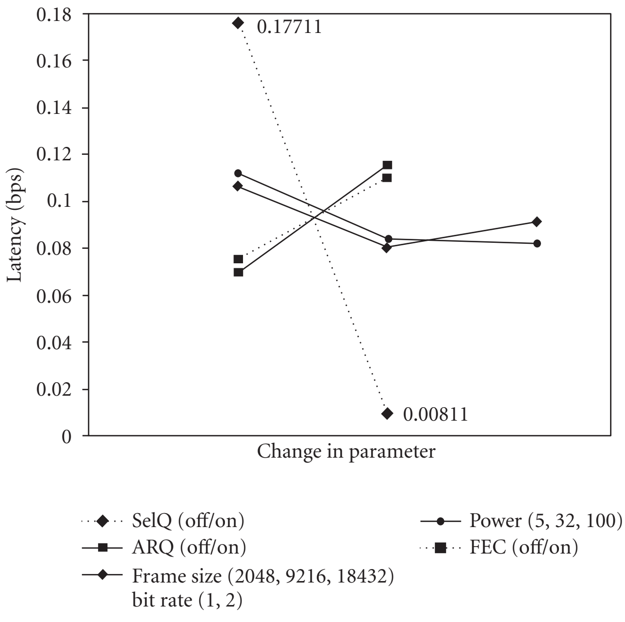

Figure 12: VoIP latency versus average effect of a parameter.

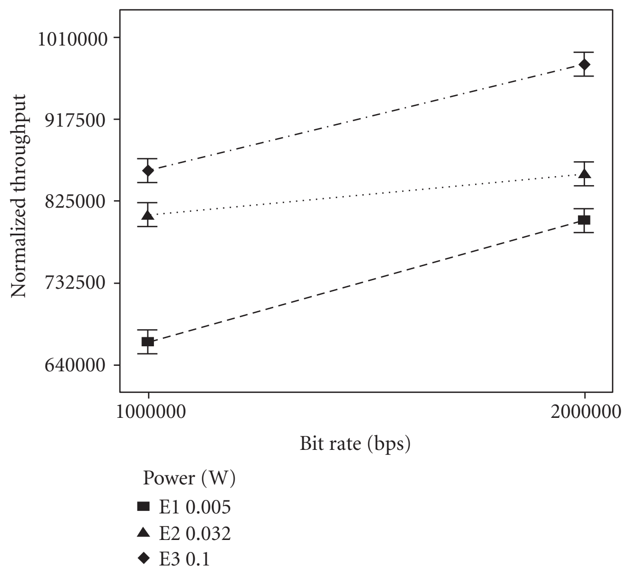

Figure 14: Analysis of power and data rate's impact on FTP throughput.

(the worst case being an average latency of 1.36 seconds and the best being 0.00937 seconds). This offers an impressive exploitation of cross-layer information yielding several orders of magnitude improvement in latency with minimal impact on FTP throughput (see Figure 11).

5.4.3. Design of experiments (DOE)

On the single-factor analysis for FTP, frame size, data rate, and power all had a positive impact, while ARQ and FEC were detrimental. More significantly, selective queueing greatly improved VoIP delay while not adversely impacting FTP throughput (as shown in Figure 13). Within the two factor analysis, we found that both ARQ and FEC had a negative effect with all parameter interactions. As shown in Figure 14, data rate demonstrated a strong synergistic effect with power.

On the single factor analysis for VoIP, Figure 15 shows that the selective queueing had the strongest impact. Likewise, data rate also demonstrated a positive effect on latency. Figure 16 shows that selective queueing on a heavily loaded channel significantly improves latency; conversely, selective queueing has no impact on a lightly loaded channel.

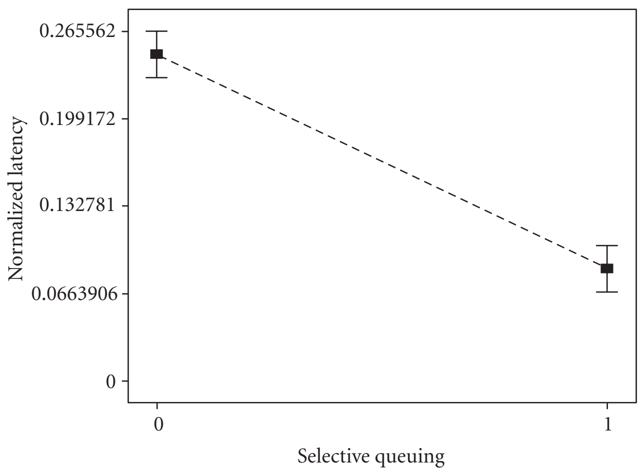

Figure 15: Analysis of selective queuing's impact on VoIP latency.

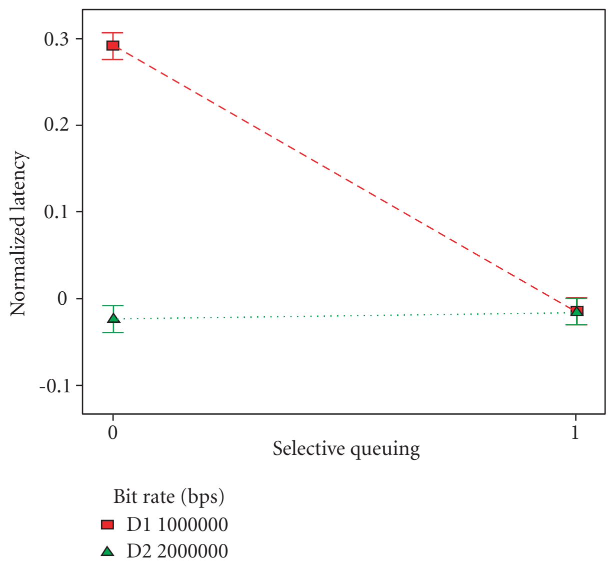

Figure 16: Analysis of selective queuing and data rate's impact on VOIP latency.

Power and selective queueing also showed a synergistic effect on improving latency. Additionally, data rate had a beneficial effect on latency regardless of power and FEC settings.

5.5. Summary

In these experiments, we were able to demonstrate a number of promising interactions that developed across parameters within the physical, data link, and network layers. Of particular significance was the ability for the platform to simultaneously support the needs of QoS-diverse applications by adapting the parameters of the system. Furthermore, this was done without increasing transmit power (while this maybe useful in improving link performance in noisy environment, it also negatively impacts neighboring nodes). In nearly all experiments, we consistently found a set of cross-layer settings that are able to support the needs of QoS-diverse applications. What makes this useful is that it shows the feasibility and promise of developing an algorithm that can adapt to the time-varying nature of environmental conditions and an active jammer.

6. AN ILLUSTRATIVE EXAMPLE

Looking forward, it is clear that there are huge performance gains to be obtained by exploiting cross-layer interactions in wireless networking. A goal of this research is the development of an operational system incorporating the full functionality of a cognitive radio. This system would autonomously select and optimize parameters according to input from the user and environment, capitalizing on cross-layer interactions as reported in this article.

To demonstrate this capability, Jeff Fifield developed an adaptive Reed-Solomon MAC that utilizes Reed-Solomon (RS) forward error correction to detect and fix bit errors in the MAC data payload. RS codes are a well-known method of encoding data for protection against transmission errors. In the RS MAC, the common $(255, 223)$ encoding scheme is used. Using this scheme, data is broken up into 223-byte blocks and each block is encoded separately, resulting in 255 bytes of encoded data. Because of the additional space and computational overhead associated with RS encoding, the MAC is adaptive, only using FEC if bit errors occur. This MAC was implemented as a click [28] application using the SoftMAC click elements and a standard RS software package. The CSMA/CA mechanism provided by SoftMAC was used for channel access. In RS MAC, all outgoing packets are either RS encoded or not. Since an endpoint cannot determine whether or not a packet it transmitted was received without error, it must rely on feedback from its peer to make transmission decisions. A simple algorithm with three configurable parameters governs the sending of these feedback packets. The parmeters are the sample period s, the error threshold e, and the no-error threshold c. Packets are observed over a sample period of s packets. If an endpoint is receiving unencoded packets and e or more packets with errors are received during a sample period, a packet is sent indicating that RS encoding should be used. Similarly, if an endpoint is currently receiving RS-encoded packets and c or more packets are received without errors during the sample period, the MAC sends a message telling its peer to stop encoding packets. In unencoded packets, errors are detected using a CRC32 checksum. In RS encoded packets, errors are detected during the RS-decoding process.

To test the functionality and performance of the adaptive Reed-Solomon MAC, we performed an experiment wherein two nodes try to send 1000-byte packets to each other at a rate of 100 packets per second. To decrease the probability of errors occurring in control frames relative to probability of errors occurring in data frames, a data rate of 1 Mbps was used for control information, while data was sent at a rate of 54 Mbps. Nodes were placed far enough apart to induce significant error when using the 54 Mbps waveform. The result of 10 trials are shown in Figure 17. For each test, 2000 packets were sent by each node for a total of 4000 packets.

Reed-Solomon MAC for $s = 10$, $e = 2$, $c = 10$				
	Recv	Valid Recv	RS Recv	Corrections
RS	3859	3660	2971	23013
No RS	3845	1850	0	0

Figure 17: Packets received, Packets correctly received, Reed-Solomon packets received, and number of Reed-Solomon corrected bytes. Averages for 10 trials of 4000 packets each.

The results show that the adaptive RS encoding scheme reduces the transmission error rate. On average, about 75 percent of packets were RS encoded, reducing the number of packets dropped due to errors from greater than 50 percent to less than 10 percent. The results also suggest that most errors occur in the 54 Mbps payload portion of the packet and not in the 1 Mbps and 2 Mbps PLCP headers. Errors were observed in more than half of the packets received, the wireless header (which cannot be disabled in SoftMAC) accounts for about 15 percent of the transmission time of a large packet for high data rates. If we assume that errors occur in half of all transmissions, that errors are equally distributed within a transmission, and that each transmission has only a single error (although the RS results suggest more than 7 errors per corrupted packet), we would expect errors in at least 7.5 percent of all headers, or in about 300 of 4000 packets. If an error occurs in the header, the frame is dropped by the hardware and the device driver never sees it. Thus, the number of packets with errors in the header is just the number of packets sent minus the number of packets received. Since the observed error rate is roughly half the predicted error rate, it must be the case that errors occur less frequently in the header than in the payload. This also validates our assumption that sending control data at 1 Mbps decreases the probability of error occuring in those frames. This simple implementation only hints at the promise of DOE as a technique for identifying beneficial cross-layer interactions (in this test, we achieved a reduction in error rate of 40 percent).

7. CONCLUSION

In this paper, we have described how parameters at the physical, data link, and network layers interact with respect to a variety of performance metrics. First, through simulation and then through experimental design techniques, we describe how parameters including power, bitrate, forward error correction, automatic repeat request, frame size, and selective queueing interact to influence throughput, bit error rate, delay and jitter. We show how such optimization can be used to improve the performance of applications by matching the settings of the lower protocol layers to the demands of the application. We then illustrate this optimization by showing how forward error correction can be used to decrease error on a noisy link by 40 percent. It is our intention to capitalize on DOE as a technique for identifying beneficial cross-layer interactions. However, the adaptive and dynamic nature of cognitive wireless systems leads to other interesting questions. It will be important to quantify the amount of time that a cognitive process can devote to computing an adaptive radio configuration, thus allowing one to characterize the types of processing that can be done without negatively affecting communication. This line of research should also provide insight into what processing should be done in real time, offline, or in the background. We plan to investigate each of these questions in future work.

ACKNOWLEDGMENTS

The authors would like to thank Eric Anderson, Gary Yee, Christian Doerr, Jeff Fifield, Mike Neufeld, and Mike Buetner for their assistance in this research.

REFERENCES

[1] J. O. Neel, J. H. Reed, and R. P. Gilles, “Game models for cognitive radio algorithm analysis,” in *Software Define Radio Forum Technical Conference (SDR ’04)*, pp. 27–32, Phoenix, Ariz, USA, November 2004.

[2] J. Neel, R. Buehrer, B. Reed, and R. P. Gilles, “Game theoretic analysis of a network of cognitive radios,” in *Proceedings of the 45th Midwest Symposium on Circuits and Systems (MWSCAS ’02)*, vol. 3, pp. 409–412, Tulsa, Okla, USA, August 2002.

[3] J. Neel, J. H. Reed, and R. P. Gilles, “Convergence of cognitive radio networks,” in *IEEE Wireless Communications and Networking Conference (WCNC ’04)*, vol. 4, pp. 2250–2255, Atlanta, Ga, USA, March 2004.

[4] C. Rieser, *Biologically inspired cognitive radio engine model utilizing distributed genetic algorithms for secure and robust wireless communications and networking*, Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Va, USA, 2004.

[5] R. I. Lackey and D. W. Upmal, “Speakeasy: the military software radio,” *IEEE Communications Magazine*, vol. 33, no. 5, pp. 56–61, 1995.

[6] R. J. Sánchez, J. B. Evans, G. J. Minden, V. S. Frost, and K. S. Shanmugan, “Rdrn: a prototype for a rapidly deployable radio network,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 2, pp. 15–22, 1998.

[7] J. Mitola III, *Cognitive radio an integrated agent architecture for software defined radio*, Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, 2000.

[8] Website, “Defense advanced research projects agency (DARPA) next generation (XG) communications,” http://www.darpa.mil/sto/smallunitops/xg.html.

[9] V. Bose, M. Ismert, M. Welborn, and J. Guttag, “Virtual radios,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 591–602, 1999, special issue on software radios.

[10] V. Bose, *Design and implementation of software radios using a general purpose processor*, Ph.D. dissertation, Massachusetts Institute of Technology (MIT), Cambridge, Mass, USA, 1999.

[11] A. Goldsmith and S. Wicker, “Design challenges for energy-constrained ad hoc wireless networks,” *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8–27, 2002.

[12] C. Barrett, A. Marathe, M. V. Marathe, and M. Drozda, “Characterizing the interaction between routing and MAC protocols in ad-hoc networks,” in *Proceedings of the 3rd ACM*

International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02), pp. 92–103, ACM Press, Lausanne, Switzerland, June 2002.

[13] R. Jiang, V. Gupta, and C. Ravishankar, "Interactions between TCP and the IEEE 802.11 MAC protocol," in *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX '03)*, vol. 1, pp. 273–282, Washington, DC, USA, April 2003.

[14] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, 2005.

[15] K. K. Vadde and V. R. Syrotiuk, "Factor interaction on service delivery in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1335–1346, 2004.

[16] K. K. Vadde and V. R. Syrotiuk, "Quantifying factors affecting quality of service in mobile ad hoc networks," *Simulation*, vol. 81, no. 8, pp. 547–560, 2005.

[17] K. K. Vadde, V. R. Syrotiuk, and D. C. Montgomery, "Optimizing protocol interaction using response surface methodology," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 627–639, 2006.

[18] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[19] L. Berlemann, S. Mangold, and B. H. Walke, "Policy-based reasoning for spectrum sharing in cognitive radio networks," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 1–10, Baltimore, Md, USA, November 2005.

[20] M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum access," in *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, pp. 78–85, Taormina, Italy, June 2005.

[21] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits in cognitive radio," in *Allerton Conference on Communication, Control and Computing*, Monticello, Va, USA, October 2004.

[22] S. Mishra, D. Cabric, C. Chang, et al., "A real time cognitive radio testbed for physical and link layer experiments," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 562–567, Baltimore, Md, USA, November 2005.

[23] R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 352–360, Baltimore, Md, USA, November 2005.

[24] C. Cordeiro, K. Challapali, D. Birru, and N. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 328–337, Baltimore, Md, USA, November 2005.

[25] Website, http://www.opnet.com/.

[26] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald, "MultiMAC—an adaptive MAC framework for dynamic radio networking," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 548–555, Baltimore, Md, USA, November 2005.

[27] M. J. Anderson and P. J. Whitcomb, *Doe Simplified: Practical Tools for Effective Experimentation*, Productivity Press, Portland, Ore, USA, 2000.

[28] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.